The image recognition training and deployment process – step by step

Most of you will be very familiar with computer images. Just like the safety belt demonstration on every commercial flight, just in case, I'll cover images, pixels, and depth. If you already know this, skip ahead.

Images are stored in a computer as a two-dimensional array of pixels, or picture elements. Each pixel is a tiny dot. Thousands or millions of tiny dots make up each image. Each individual pixel is a number or series of numbers that describe its color. If the image is only a gray-scale, or black and white image, then each pixel is represented by a single number that corresponds to how dark or light the tiny dot is. This is fairly straightforward so far.

If the image is a color picture, then each dot has three numbers that are combined to make its color. Usually, these numbers are intensity of Red, Green, and Blue (RGB) colors. The combination (0,0,0) is black (or the absence of all color), and (255,255,255) is white – the sum of all colors. This process is called the additive color model. If you work with watercolors instead of computer pixels, you know that adding all the colors in your watercolor box makes black – that is, a subtractive color model.

While RGB is one set of three numbers that can describe a pixel, there are other ways of describing the "color formula" that have various usages. We don't have to use RGB, for instance; we can also use CYM – Cyan, Yellow, and Magenta, which are the complementary colors to Red, Green, and Blue. We can also break down colors using the HSV (Hue, Saturation, and Value) model, which classifies color by Hue (type of color), Saturation (intensity of color), and Value, or brightness of color. HSV is a very useful color space for certain calculations, such as converting a color image to gray-scale (black and white). To turn an RGB to a gray-scale pixel, you have to do a bit of math – you can't just pull out one channel and keep it. The formula for RGB to gray-scale is (*0.3*Red + 0.58*Green + 0.11*Blue*). This is because the different wavelengths of light behave differently in our eyes, which are more sensitive to green. If you have color in the HSV color model, then creating a grayscale image is just taking the V(value) number and throwing the Hue and Saturation away. You can see that is a lot simpler.

We will be doing quite a bit of image manipulation through this chapter.

An RGB pixel is represented by the three colors, which means the actual image is a three-dimensional array rather than two-dimensional, because each pixel has three numbers, making an array of (height, width, 3). So, a picture that is 800 by 600 would have an array of (800,600,3), or 1,440,000 numbers. That is a lot of numbers. We will be working very hard to minimize the numbers of pixels we are processing at any given time.